

BAB II

LANDASAN TEORI

2.1 Tinjauan jurnal

Kebugaran jasmani merupakan tuntutan dalam hidup agar lebih sehat dan mampu beraktivitas secara produktif. Kebugaran jasmani menjadi bagian dari program pendidikan, pembinaan kebugaran jasmani sangat strategis, karena mendukung dan dapat memberikan semangat bagi partisipasi pengguna secara menyeluruh, (Rusli Lutan, 2012: 1). Menurut Giri Wiarto (2015: 55) Kebugaran jasmani adalah kesanggupan dan kemampuan tubuh untuk melakukan penyesuaian terhadap pembebasan fisik yang diberikan tanpa menimbulkan kelelahan yang berlebihan. Secara umum yang dimaksud dengan kebugaran jasmani adalah kebugaran fisik (*physical fitness*), yaitu kemampuan seseorang untuk melakukan kerja sehari-hari secara efisien tanpa timbul kelelahan yang berlebihan, sehingga dapat menikmati waktu luangnya (Djoko Pekik Irianto, 2012:2)

Masyarakat pada umumnya mengenal kebugaran jasmani dengan istilah *physical fitness*. Menurut Santoso Giriwijoyo dan Didik Zafar (2012:17) *Physical fitness* selain diterjemahkan sebagai kebugaran jasmani, diterjemahkan pula dengan istilah-istilah lain misalnya: (1) Kesegaran Jasmani, (2) Kesanggupan Jasmani, dan (3) Kesamaptaan Jasmani.

Berdasarkan beberapa pendapat di atas maka dapat disimpulkan bahwa kebugaran jasmani adalah kemampuan seseorang menjalankan berbagai macam aktivitas

sehari – hari dan masih mempunyai tenaga cadangan untuk menikmati waktu senggang maupun melakukan keperluan yang lain

Kebugaran jasmani memiliki beberapa komponen yang saling berkaitan antara satu dengan yang lain. Namun, masing-masing komponen memiliki ciri – ciri yang berfungsi pokok dalam kebugaran jasmani seseorang. Kebugaran jasmani sangat penting dalam menunjang aktivitas kehidupan sehari-hari, akan tetapi nilai kebugaran jasmani tiap-tiap orang berbeda-beda sesuai dengan tugas/profesi masing-masing. Kebugaran jasmani terdiri dari komponen-komponen yang dapat dikelompokkan menjadi kelompok kebugaran yang berhubungan dengan kesehatan (*health related physical fitness*) dan kelompok kebugaran yang berhubungan dengan keterampilan (*Skill related physical fitness*). Menurut Giri Wiarto (2015:58) kriteria kebugaran jasmani ditentukan oleh 2 komponen yaitu :

1. Komponen kesehatan

Komponen kesehatan yang berpengaruh terhadap kebugaran jasmani adalah daya tahan kardiovaskuler, daya tahan dan kekuatan otot, komposisi tubuh dan fleksibilitas

2. Komponen keterampilan

Komponen keterampilan yang berpengaruh dalam kebugaran adalah kekuatan otot, kelincahan, kecepatan, bentuk tubuh, dan ketebalan otot.

Komponen-komponen kebugaran jasmani tersebut dapat dijelaskan sebagai berikut:

- A. Komponen Kebugaran Berkaitan dengan Kesehatan

- a) Komposisi tubuh

Komposisi tubuh adalah persentase (%) lemak dari berat badan total dan Indeks Masa Tubuh (IMT) (Giri Wiarto, 2015:56). Menurut Djoko Pekik (2012:4) Komposisi tubuh adalah perbandingan berat tubuh berupa lemak dengan berat tubuh tanpa lemak yang dinyatakan dalam persentase lemak tubuh.

Komposisi tubuh dapat diukur dengan skinfold caliper dan IMT (Indeks Masa Tubuh). Pengukuran menggunakan skinfold caliper dilakukan pada bagian tubuh yaitu bagian dada, subscapula (lipatan kulit di atas tulang belikat), mid-axilla (tengah-tengah antara tulang pinggul dan ketiak), suprailiaca (pinggang/ diatas tulang panggul), dan bagian abdomen (perut), sedangkan besarnya Indeks Massa Tubuh dapat diketahui menggunakan formulasi sebagai berikut :

Tabel II.1 Kriteria indeks Masa Tubuh

$$IMT = \text{Berat Badan (Kg)} / \text{Tinggi Badan (M}^2\text{)}$$

Hasil perhitungan IMT dikonsultasikan dengan tabel berikut:

Tabel 1. Kriteria Indeks Masa Tubuh

Status Gizi	Laki-Laki	Perempuan
Kurus	< 20.1	<18.7
Normal	20.1 – 25.0	18.7 – 23.8
Obsese	>30	>28.6
Rata-rata	22.0	20.8

(Sumber: Djoko Pekik, 2007: 74)

b) Kelenturan (*Fleksibility*)

Kelenturan adalah kemampuan persendian bergerak secara leluasa

(Djoko Pekik, 2012: 4). Peter J.L. Thomson (2015: 75) menerangkan bahwa fleksibilitas/kelenturan adalah kemampuan melakukan gerakan persendian melalui jangkauan gerak yang luas. Menurut Giri Wiarto (2015: 56) Kelenturan adalah luas bidang gerak yang maksimal pada persendian, tanpa dipengaruhi oleh suatu paksaan atau tekanan.

Puncak kelenturan terjadi pada akhir masa pubertas. Fleksibilitas ini sangat penting pada setiap gerak tubuh karena meningkatkan efisiensi kerja otot. Fungsi dari fleksibilitas tubuh adalah dapat mengurangi cedera (Giri Wiarto, 2015: 56).

Berdasarkan beberapa pendapat di atas, fleksibilitas /kelenturan /kelenturan adalah jangkauan terluas yang dapat dilakukan oleh persendian tanpa adanya paksaan. Contoh latihan kelenturan adalah latihan peregangan baik statis maupun dinamis, dan peregangan pasif.

c) Kekuatan Otot

Kekuatan otot adalah kemampuan badan dalam menggunakan daya, serabut otot yang ada dalam otot akan memberikan tanggapan apabila dikenakan beban atau tahanan dalam latihan (Peter J.L. Thomson, 2015: 70). Menurut Giri Wiarto (2015:55) kekuatan otot adalah kontraksi maksimal yang dihasilkan otot, merupakan kemampuan untuk membangkitkan tegangan terhadap suatu tegangan. Djoko Pekik (2012:4) menerangkan bahwa kekuatan otot adalah kemampuan otot melawan beban dalam suatu usaha.

Kekuatan maksimum otot yang terbaik dikembangkan oleh latihan-latihan yang melibatkan pengulangan dalam jumlah yang sedikit dengan

tahanan yang besar. Menurut Peter J.L dalam latihan kekuatan terdapat istilah-istilah yang digunakan, yaitu : (1) Tahanan (resistance) adalah suatu beban yang diminta untuk di gerakan oleh sebuah atau suatu kelompok otot, (2) Pengulangan (repetition) adalah jumlah berapa kali satu latihan dilakukan tanpa henti, dan (3) set/pasang adalah satu jumlah pengulangan yang ditentukan merupakan satu set cara menulisnya untuk 3 set 10 pengulangan adalah 3 x 10 x tahanan.

Berdasarkan beberapa pendapat di atas dapat disimpulkan kekuatan otot adalah kemampuan satu atau kelompok otot dalam tubuh untuk mengarahkan daya dalam melakukan tugas gerak dengan usaha maksimum. Contoh latihan untuk kekuatan otot adalah angkat beban.

d) Daya Tahan Otot

Daya tahan otot adalah kemampuan sekelompok otot untuk mengerahkan daya maksimum selama periode waktu yang relatif lama terhadap sebuah tahanan yang lebih ringan dari pada beban yang bisa digerakkan oleh seseorang (Rusli Lutan, 2013: 56). Menurut Giri Wiarto (2015: 58) daya tahan otot merupakan kemampuan untuk kontraksi sub-maksimal secara berulang-ulang atau berkontraksi terus menerus dalam suatu waktu tertentu.

Berdasarkan beberapa pengertian di atas dapat disimpulkan bahwa daya tahan otot adalah kemampuan satu atau kelompok otot dalam tubuh untuk mengerahkan daya dalam satu periode waktu terhadap beban yang

dapat digerakkan. Contoh latihan untuk daya tahan otot adalah push up, sit up, dan naik turun bangku.

e) Daya Tahan Jantung Paru

Aerobik berarti menggunakan oksigen. Menurut Peter J.L. Thomson (2015: 72) Daya tahan aerobik berarti kerja otot dan gerakan otot yang dilakukan menggunakan oksigen guna melepaskan energi dari bahan-bahan otot. Latihan aerobik menuntun untuk memperkuat sistem kardiorespiratori dan suatu peningkatan kemampuan dalam menggunakan oksigen di dalam otot. Menurut Djoko Pekik (2012: 4) Daya Tahan Jantung Paru yaitu kemampuan paru-paru jantung mensuplai oksigen untuk kerja otot dalam jangka waktu yang lama. Giri Wiarto (2015: 58) menyatakan bahwa daya tahan jantung paru adalah kemampuan jantung, paru dan pembuluh darah untuk berfungsi secara optimal pada waktu kerja dalam mengambil O₂ secara maksimal dan menyalurkannya ke seluruh tubuh untuk proses metabolisme tubuh.

Berdasarkan pendapat-pendapat di atas dapat disimpulkan bahwa daya tahan aerobik/daya tahan paru jantung adalah kemampuan paru-paru dan jantung dalam menyuplai oksigen yang digunakan dalam kerja otot pada jangka waktu yang lama. Latihan untuk meningkatkan daya tahan jantung paru adalah lari secara terus menerus, lari dengan kecepatan bervariasi dan lari fartlek.

2) Komponen Kebugaran Berkaitan dengan Keterampilan

Komponen-komponen kebugaran jasmani yang berkaitan dengan keterampilan dapat dijelaskan sebagai berikut (Giri Wiarto, 2015: 57-58):

a) Kelincahan

Kelincahan adalah kemampuan seseorang untuk dapat mengubah arah dengan cepat dan tepat pada waktu bergerak tanpa kehilangan keseimbangan. Beberapa contoh latihan kelincahan adalah lari melewati tiang penghalang, shuttle run training.

b) Kekuatan Otot

Kekuatan otot adalah kontraksi maksimal yang dihasilkan otot, merupakan kemampuan untuk membangkitkan tegangan terhadap suatu tegangan.

c) Kecepatan

Kecepatan adalah kemampuan menempuh jarak tertentu dalam waktu seminimal mungkin. Contoh untuk latihan kecepatan adalah lari cepat dengan jarak 50 meter, 60 meter.

Komponen-komponen kebugaran memiliki dua kriteria yaitu komponen kebugaran yang berkaitan dengan kesehatan dan komponen kebugaran yang berkaitan dengan keterampilan. Komponen kebugaran yang berkaitan dengan kesehatan adalah komposisi tubuh, fleksibilitas, daya tahan otot, kekuatan otot, dan daya tahan jantung-paru. Sedangkan komponen kebugaran yang

berkaitan dengan keterampilan adalah kekuatan otot, kelincahan, kecepatan, bentuk tubuh, dan ketebalan otot.

B. Manfaat Memiliki Kebugaran Jasmani

Gerak dan aktivitas fisik merupakan fungsi dasar untuk semua organisme hidup. Sehat, vitalitas dan panjang umur merupakan harapan semua orang, namun semua itu tidak akan pernah diperoleh jika tanpa diikuti oleh usaha yang memadai.

Menurut Giri Wiarto (2015: 59) latihan fisik memiliki peranan penting untuk mempertahankan dan meningkatkan derajat kebugaran seseorang. Derajat jasmani seseorang sangat menentukan kemampuan fisiknya dalam melaksanakan tugas sehari-hari. Semakin tinggi derajat kebugaran jasmani seseorang, semakin tinggi pula kemampuan kerja fisiknya. Selain itu memiliki kebugaran jasmani dapat mencegah terserang penyakit jantung, stroke, hipertensi, dan osteoporosis.

Olahraga memiliki banyak Fungsi bagi kesehatan tubuh baik kesehatan jasmani maupun kesehatan rohani, berikut penjelasan mengenai manfaat olahraga terhadap kesehatan jasmani dan rohani (Giri Wiarto, 2015: 16-23):

C. Prinsip – Prinsip Kebugaran Jasmani

Aktivitas kebugaran jasmani yang akan dilakukan tidak dapat dilakukan dengan sembarang. Latihan kebugaran jasmani harus memperhatikan prinsip-prinsip baik dalam melakukan latihan maupun saat merencanakan program latihan. Latihan kebugaran jasmani merupakan salah satu usaha untuk menyesuaikan keadaan jasmani terhadap kegiatan yang diperberat atau lebih berat dengan cara bertahap sehingga akan mengalami adaptasi dalam menerima atau melakukan kegiatan yang lebih berat. Latihan kebugaran diartikan sebagai proses sistematis menggunakan gerakan yang bertujuan meningkatkan atau mempertahankan kualitas fungsi tubuh meliputi kualitas daya tahan paru jantung, kekuatan dan daya tahan otot, kelentukan, dan komposisi tubuh (Djoko Pekik, 2004: 12). Prinsip-prinsip latihan kebugaran menurut Djoko Pekik (2004: 12), yaitu sebagai berikut : (1) Beban lebih (Overload), (2) Kekhususan (Specificity), dan (3) Kembali asal (Reversible).

2.2 Konsep Dasar Program

Pengertian pemrograman berorientasi objek menurut Rosa (2014:100) “Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya”. Aktivitas kebugaran yang akan dilakukan tidak dapat dilakukan dengan sembarang. Latihan kebugaran harus memperhatikan prinsip-prinsip baik dalam melakukan latihan maupun saat merencanakan program latihan. Latihan kebugaran merupakan salah satu usaha untuk menyesuaikan keadaan terhadap kegiatan yang diperberat atau lebih berat

dengan cara bertahap sehingga akan mengalami adaptasi dalam menerima atau melakukan kegiatan yang lebih berat. Latihan kebugaran diartikan sebagai proses sistematis menggunakan gerakan yang bertujuan meningkatkan atau mempertahankan kualitas fungsi tubuh meliputi kualitas daya tahan paru jantung, kekuatan dan daya tahan otot, kelentukan, dan komposisi tubuh (Djoko Pekik, 2011: 12). Prinsip-prinsip latihan kebugaran menurut Djoko Pekik (2011: 12), yaitu sebagai berikut : (1) Beban lebih (*Overload*), (2) Kekhususan (*Specificity*), dan (3) Kembali asal (*Reversible*).

Prinsip – prinsip dalam latihan kebugaran memiliki pengertian sebagai berikut:

A. Konsep Dasar Program Berorientasi Objek

1) *Overload*

Pembebanan yang diberikan dalam latihan harus lebih berat dibandingkan aktivitas fisik sehari-hari yang biasa dilakukan (Djoko Pekik, 2011: 12). Menurut Giri Wiarto (2015: 33) Prinsip beban berlebih yaitu dalam setiap melakukan aktivitas fisik harus selalu diupayakan adanya penambahan beban latihan antara latihan saat itu dan latihan selanjutnya. Misalnya: seseorang yang setiap berangkat kerja berjalan sejauh 500 meter, maka pada saat berlatih untuk meningkatkan kebugarannya, harus menempuh jarak yang lebih jauh atau berjalan lebih cepat.

Berdasarkan beberapa pendapat di atas prinsip beban lebih (*Overload*) dapat disimpulkan bahwa dalam memberikan beban latihan adalah lebih berat dari

beban yang diterima dalam aktivitas sehari-hari dan ada penambahan beban dalam latihan selanjutnya.

1) *Specificity*

Latihan yang dipilih harus sesuai dengan tujuan latihan yang hendak dicapai (Djoko Pekik, 2011: 12). Menurut Giri Wiarto (2015: 33) bentuk latihan hendaknya bersifat spesifik sesuai dengan maksud dan tujuan latihan dilakukan. Misalnya: untuk menurunkan berat badan pilihlah latihan aerobik, sedangkan untuk melatih kekuatan otot pilihlah latihan berat.

Berdasarkan pendapat-pendapat di atas kekhususan pada prinsip-prinsip latihan kebugaran dapat dikatakan sebagai penyesuaian bentuk latihan dengan tujuan dan maksud dari latihan yang akan dicapai.

2) *Reversible*

Kebugaran yang telah dicapai akan berangsur-angsur menurun bahkan bisa hilang sama sekali jika latihan tidak dikerjakan secara teratur dengan takaran yang tepat. Kebugaran akan menurun 50% setelah berhenti latihan 4-12 Minggu dan akan terus berkurang hingga 100% setelah 10-30 Minggu (Djoko Pekik, 2011:12). Prinsip kembali asal dapat dijelaskan sebagai berikut: latihan yang dilakukan telah menyesuaikan dengan beban yang diterima dan akan terus meningkat seiring dengan prinsip *overload*. Bila latihan tidak dilakukan secara teratur atau terhenti selama beberapa waktu maka kemampuan untuk menyesuaikan beban yang

diterima akan menurun bahkan menghilang dan kembali ke keadaan semula yaitu keadaan awal sebelum melakukan latihan.

Latihan dalam meningkatkan kebugaran juga dipengaruhi oleh takaran atau dosis latihan. Menurut Djoko Pekik Irianto (2011: 14), keberhasilan dalam meningkatkan kebugaran sangat dipengaruhi oleh takaran atau dosis latihan yang dijabarkan dalam konsep FITT (Frekuensi, Intensitas, *Time*, Tipe). Adapun pengertian dari FITT adalah :

B. Konsep Dasar Program Android

1) *Frekuensi*

Frekuensi adalah banyaknya unit latihan per Minggu. Untuk meningkatkan kebugaran perlu latihan 3-5 kali per Minggu. Sebaiknya dilakukan berselang, misalnya Senin, Rabu, Jumat. Sedangkan hari yang lain digunakan untuk istirahat agar tubuh memiliki kesempatan melakukan *recovery* (pemulihan) tenaga (Djoko Pekik Irianto, 2011: 14). Menambahkan, menurut Giri Wiarto (2015: 33) frekuensi adalah jumlah latihan yang dilakukan dalam periode waktu tertentu.

Berdasarkan pendapat di atas frekuensi latihan adalah banyaknya latihan yang dilakukan pada periode tertentu.

2) *Intensitas*

Intensitas adalah kualitas yang menunjukkan berat ringannya latihan. Besarnya intensitas tergantung pada jenis dan tujuan latihan. Secara umum intensitas

latihan kebugaran adalah 60 %-90 % detak jantung maksimal dan pembakaran lemak 65 %-75 % detak jantung maksimal. Latihan daya tahan paru-jantung 75 %-85 % detak jantung maksimal dan latihan anaerobik untuk atlet > 85 % detak jantung maksimal (Djoko Pekik Irianto, 2011: 14).

Intensitas adalah ukuran untuk menunjukkan kualitas suatu rangsangan atau pembebanan. Untuk menentukan besarnya intensitas latihan dapat diketahui salah satunya adalah dengan cara mengetahui denyut nadi maksimal. Rumus untuk menentukan denyut nadi maksimal adalah Denyut Nadi Maksimal = 220 - umur (Giri Wiarto, 2015: 33).

Berdasarkan pendapat di atas dapat disimpulkan bahwa intensitas adalah ukuran yang menunjukkan kualitas suatu latihan yang dapat ditentukan dengan mengetahui target dari Denyut Nadi Maksimal yang harus dicapai.

3) *Time*

Time adalah waktu atau durasi yang diperlukan setiap kali berlatih. Untuk meningkatkan kebugaran paru-jantung dan penurunan berat badan diperlukan waktu berlatih 20-30 menit, dan hasilnya akan tampak nyata setelah berlatih 8-12 Minggu dan akan stabil setelah 20 Minggu berlatih (Djoko Pekik, 2011: 14). Giri Wiarto (2015: 33) mengatakan bahwa lamanya latihan adalah ukuran yang menunjukkan lama waktu latihan dalam setiap melakukan latihan disebut juga durasi latihan. Berdasarkan pendapat di atas dapat disimpulkan bahwa *Time* dalam FITT adalah ukuran lamanya waktu atau durasi dalam setiap latihan.

4) *Tipe*

Tipe adalah sebuah bentuk atau model latihan yang akan digunakan untuk mencapai sasaran latihan tertentu (Djoko Pekik, 2011: 14). Bentuk latihan hendaknya bersifat spesifik sesuai dengan maksud dan tujuan latihan dilakukan (Giri Wiarto, 2015:33). Berdasarkan pendapat di atas dapat disimpulkan bahwa tipe adalah sebuah model latihan yang digunakan berdasarkan tujuan dan sasaran dalam melakukan latihan.

2.3. Metode Algoritma

Kata *algorism* berasal dari nama penulis buku Arab yang terkenal, yaitu Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi (al-Khuwarizmi dibaca orang Barat menjadi *algorism*). Dalam bahasa Indonesia, kata *algorism* diserap menjadi algoritma.

Menurut Ananda (2009:3) “Kata ‘algoritma’ diturunkan dari nama belakang seorang tokoh metematikawan Persia bernama Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi (lahir tahun 730an, meninggal antara tahun 835 dan 850).”

Al-Khuwarizmi berasal dari propinsi Khorasan di negara yang kini bernama Uzbekistan. Uni Soviet (Rusia) menghormati jasa-jasa Al-Khuwarizmi dengan membuat gambar dirinya sebagai perangko.

Menurut Munir (2007:4) “Algoritma adalah urutan langkah-langkah untuk memecahkan suatu masalah”. Terdapat beberapa definisi lain dari algoritma, tetapi pada perinsipnya senada dengan definisi yang diungkapkan diatas.

Pemrograman sudah menjadi kegiatan yang sangat penting di era teknologi informasi. Program yang berjalan di komputer *desktop*, *laptop*, telepon *genggam*, *PDA*, dan sebagainya, tidak tercipta begitu saja, tetapi ditulis melalui proses analisis dan perancangan yang cermat. Sebuah program komputer pada dasarnya mengimplementasikan suatu algoritma. Algoritma akan lebih baik jika ditulis secara matematis menggunakan skema, metode algoritma yang akan dituangkan pada penulisan ini dalam bentuk algoritma pemrograman java (*Flowchart*) dan *Pseudocode*.

A. Euclid's Algorithm

Pada tahun 1950, kata algoritma pertama kali digunakan pada algoritma *Euclidean*, *Euclid* seorang matematikawan Yunani (lahir pada tahun 350 M), dalam bukunya yang berjudul *Element* menuliskan langkah-langkah untuk menemukan pembagi bersama terbesar (*common greatest divisor* atau *gcd*), dari dua buah bilangan bulat, m dan n . Pembagi bersama terbesar dari dua buah bilangan bulat tak negatif merupakan bilangan bulat positif terbesar yang habis membagi kedua bilangan tersebut.

Misalnya, $m = 80$ dan $n = 12$. Semua faktor pembagi 80 adalah,

1, 2, 4, 5, 8, 10, 16, 20, 40, 80,

Dan semua faktor pembagi 12 adalah,

1, 2, 3, 4, 6, 12,

Maka $gcd(80,12) = 4$. Langkah-langkah mencari $gcd(80,12)$ dengan algoritma *Euclidean* sebagai berikut.

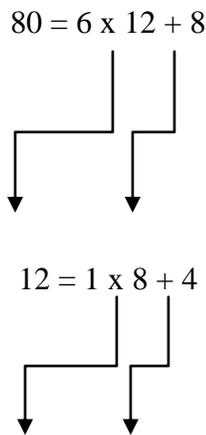
80 dibagi 12 hasilnya = 6, sisa = 8 (atau: $80 = 6 \times 12 + 8$)

12 dibagi 8 hasilnya = 1, sisa = 4 (atau: $12 = 1 \times 8 + 4$)

8 dibagi 4 hasilnya = 2, sisa = 0 (atau: $8 = 4 \times 2 + 0$)

Karena pembagian yang terakhir menghasilkan 0, maka sisa pembagian terakhir sebelum 0, yaitu 4, menjadi $gcd(80,12)$.

Jadi, $gcd(80,12) = gcd(12,8) = gcd(8,4) = gcd(4,0) = 4$. Proses mencari gcd dari 80 dan 12 juga dapat diilustrasikan dalam diagram berikut:



Terdapat beberapa versi algoritma *Euclidean*, salah satu contoh versinya dituliskan

Algoritma *Euclidean* sebagai berikut.

{ Diberikan dua buah bilangan bulat tak-negatif m dan n ($m \geq n$). Algoritma *Euclidean* mencari pembagi bersama terbesar, gcd dari kedua bilangan tersebut, yaitu bilangan bulat positif terbesar yang habis membagi m dan n . }

1. Jika $n = 0$, maka m adalah jawabannya.

Stop.

Tetapi jika $n \neq 0$, lanjutkan langkah 2.

2. Bagilah m dengan n dan misalkan r adalah sisanya,

3. Ganti nilai m dengan nilai n , dan nilai n dengan nilai r , lalu ulang kembali ke langkah 1.

Dari satu contoh tersebut diatas bahwa metoda atau cara dan teknik algoritma pemrograman dapat diterapkan dengan berbagai metode yang terpenting memenuhi struktur (konstruksi) dasar algoritma yaitu, Runtunan, Pemilihan, dan Pengulangan. Runtunan merupakan deretan instruksi yang dikerjakan sesuai urutan penulisannya. Pemilihan berisi struktur program yang memiliki satu atau lebih pilihan aksi yang dilakukan, notasi *if*, *if-then*, *if-then-else*, dan *switch case* digunakan untuk menyatakan konstruksi pemilihan. Pengulangan berisi struktur pengulangan yang didalamnya berisi satu atau lebih aksi yang dikerjakan secara berulang-ulang, notasi *for*, *while* dan *repeat-until* digunakan untuk menyatakan konstruksi perulangan.

B. *Flowchart* (Diagram Alur)

Menurut Ananda (2009:8) “*Flowchart* didefinisikan sebagai skema penggambaran dari algoritma atau proses.” Metode untuk menggambarkan tahap-tahap pemecahan masalah dengan menggambarkan simbol-simbol tertentu yang mudah dimengerti, mudah digunakan dan standart. Dalam membuat algoritma, diperlukan suatu mekanisme atau pola diagram untuk menuangkan hasil pemikiran mengenai langkah-langkah penyelesaian masalah yang sistematis dan terurut susunan logika suatu program memakai simbol-simbol sebagai berikut.

Tabel II.2 Simbol-simbol dalam *flowchart*

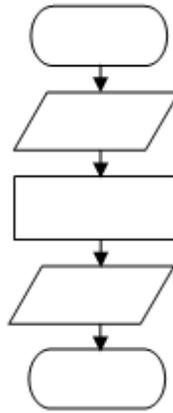
	Terminator Sebagai simbol 'Start' dan 'End' untuk memulai dan mengakhiri <i>flowchart</i>
	Input/Output Digunakan untuk menuliskan proses menerima data atau mengeluarkan data
	Process Digunakan untuk menulis proses yang diperlukan, misalnya operasi <i>aritmatika</i>
	Conditional/Decision Digunakan untuk menyatakan proses yang membutuhkan keputusan
	Preparation Digunakan untuk memberikan nilai awal
	Connector (on-page) Digunakan untuk menyatukan beberapa <i>arrow</i> (anak panah).
	Connector (off-page) Digunakan untuk menghubungkan <i>flowchar</i> pada halaman berikutnya
	Display Digunakan untuk menampilkan data ke monitor atau layar
	Arrow Sebagai petunjuk arah dan alur proses

Sumber: Algoritma dan pemrograman (2009:8)

Folwchart tersebut terdiri dari tiga alur data yaitu pembacaan atau pemasukan data (*input*), komputasi perhitungan terhadap data tersebut (*process*) dan menampilkan hasilnya (*output*). Struktur standart terdiri dari tiga komponen sebagai berikut.

1. Struktur *Squence* (Linear)

Diagram yang alurnya mengalir secara berurutan dari atas kebawah atau dengan kata lain tidak adanya percabangan atau pengulangan.

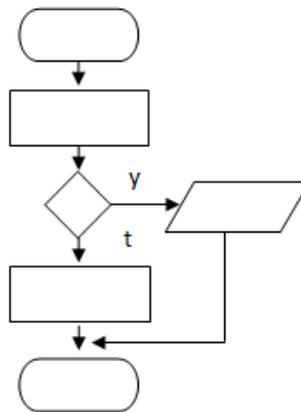


Gambar II.1 *Flowchart* struktur *sequence*

Sumber: Ananda (2009:9)

2. Struktur *Branching* (Bercabang)

Diagram ini terjadi atau terdapat alih kontrol berupa percabangan, digunakan untuk meyeleksi kondisi dan menentukan pilihan proses selanjutnya.

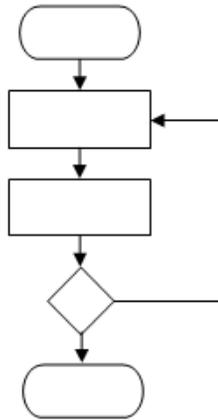


Gambar II.2 *Flowchart* struktur *branching*

Sumber: Ananda (2009:15)

3. Struktur Looping (Perulangan)

Flowchart dengan struktur perulangan dan digunakan untuk mengulangi langkah-langkah sebelumnya sampai kondisi terpenuhi.



Gambar II.3 *Flowchart* struktur *looping*

Sumber: Ananda (2009:17)

C. *Pseudocode*

Menurut Barakbah (2013:17) “*Pseudocode* adalah cara penulisan algoritma yang menyerupai bahasa pemrograman tingkat tinggi.” *Pseudocode* menggunakan bahasa yang hampir menyerupai bahasa pemrograman. Biasanya *pseudo-code* menggunakan bahasa yang mudah dipahami secara universal dan juga lebih ringkas dari pada algoritma. *Pseudocode* berisi deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dari beberapa bahasa pemrograman tetapi bahasa tersebut hanya di tujukan agar dapat di baca manusia. Sehingga *pseudocode* tidak dapat dipahami oleh komputer. Supaya notasi *pseudocode* bisa dipahami oleh komputer maka harus diterjemahkan terlebih dahulu menjadi *sintaks* bahasa pemrograman komputer tertentu. Dalam *pseudocode*, tidak ada sintaks standar yang

resmi. Karena itu, *pseudocode* ini dapat diterapkan dalam berbagai bahasa pemrograman. Disarankan untuk menggunakan *keyword* yang umum digunakan seperti : *if, then, else, while, do, repeat, for*, dan lainnya. Keuntungan menggunakan notasi *pseudocode* kemudahan mentranslasi ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap *pseudocode* dengan notasi bahasa pemrograman. Perbandingan beberapa kata atau kalimat yang biasa digunakan dalam penulisan algoritma dengan menggunakan kalimat deskriptif dan *pseudocode*. ar Program Android

Konsep dasar pemrograman aplikasi android ditulis dalam bahasa pemrograman *java*. Android merupakan *platform* pertama yang lengkap, terbuka dan bebas.

Menurut Safaat (2015:9) Kode java dikompilasi bersama-sama dengan data file *resource* yang dibutuhkan oleh aplikasi, dimana prosesnya di *packege* oleh *tools* yang dinamakan “*apt tools*” kedalam paket android sehingga menghasilkan file dengan ekstensi apk, file apk itulah yang kita sebut aplikasi, dan nantinya dapat di install di perangkat *mobile*.

Ada empat jenis pokok komponen dalam merancang aplikasi pemrograman pada aplikasi berbasis android yaitu:

1. *Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan intraksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity*

biasanya akan dipakai untuk menampilkan aplikasi atau bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain dapat dilakukan dengan satu even, misalnya *click* tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method Activity setContentView()*. *ContentView* merupakan objek yang berada pada *root* hirarki.

a. Konsep *Activity*

Pada pemrograman lain yang meluncurkan aplikasi dengan metode *main()*, sistem Android memulai kode dalam *instance Activity* dengan memanggil metode *callback* tertentu yang menyatakan tahap tertentu dari daur hidupnya. Ada urutan metode *callback* dimulainya aktivitas dan urutan metode *callback* hingga menghentikan aktivitas. *Activity* merupakan *public class* dalam aplikasi android, setiap *activity* merupakan suatu yang unik atau *single*, yang ditujukan untuk menhandel macam-macam hal yang bisa dilakukan oleh *user*. Umumnya *activity* berhubungan dengan *user* dimana *activity* menciptakan *windows* atau UI yang mana ditampilkan dengan *concepts setContentView(View)*. Ada 2 method yang dimiliki oleh satu *activity*, yaitu:

- 1) *onCreate* untuk menginisiasi suatu *activity*, biasanya dipanggil dengan perintah *setContentView(int)* untuk *resource* yang didefinisikan di *layout* UI, dan perintah *findViewById(int)* untuk memanggil *widget* yang dibutuhkan *user interface* untuk berinteraksi dengan aplikasi.
- 2) *onPause* untuk menyatakan ketika *user* meninggalkan suatu *activity*. Penggunaan dengan *Context.startActivity()*, semua kelas *Activity* harus

sesuai dengan `<activity>` yang di deklarasikan satu paket di *AndroidManifest.xml*.

b. Konsep *Widget*

Widget merupakan visualisasi dari elemen *user interface* (UI) yang digunakan pada layar aplikasi android dimana kita dapat merancang sesuai dengan kebutuhan. *Widget* dalam android ditampilkan dengan konsep *View*. Aplikasi android pada umumnya menggunakan *widget* sebagai *Layout.xml*. Untuk mengimplementasikan *widget*, selain file *java* dibutuhkan dua file tambahan.

1) *File Java*

Berupa file *java* merupakan file yang mengimplementasikan aksi dari *widget*. Jika mendefinisikan *widget* dalam pemrograman, maka posisinya dilayar yang didefinisikan dari file *XML* (*extended markup language*). Coding di file *java* dapat mengambil semua nilai atribut dari tata letak file *XML* yang didefinisikan. Sintaks penulisannya

2) *File XML (Extended Markup Language)*

Sebuah file *XML* yang berada di *res/values/* mendefinisikan komponen element-element *XML*. Digunakan untuk inisialisasi *widget* serta atribut yang mendukungnya, file ini dikenal dengan nama *main.xml*

3) *Layout XML* [optional]

File ini tidak harus ada, umumnya digunakan dan sangat membantu serta dapat memudahkan didalam penyusunan pemrograman android. File *layout.xml* berada didalam *res/layout/* yang menggambarkan atau

penambahan keterangan pada layout *widget*, file ini dikenal dengan *string.xml*

2. *Service*

Service tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara background, misalnya dalam memainkan musik, *service* mungkin saja mengambil data dari jaringan, tetapi *service* harus berada dalam induknya. Misalnya pada media *player* sedang memutar lagu dari list yang ada atau menulis sms sambil *player* sedang berjalan. Untuk menjaga musik tetap berjalan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada thread utama dari proses aplikasi

3. *Broadcast Receiver*

Broad receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Misalnya notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh kamera, atau perubahan referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast*, memberikan informasi pada aplikasi lain bahwa ada data yang telah diunduh ke perangkat dan siap untuk digunakan. *Broad receiver* tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima, atau menggunakan notifikasi manager untuk memberi tahu pengguna seperti lampu latar, *vibrating* (getaran) perangkat.

4. *Content Provider*

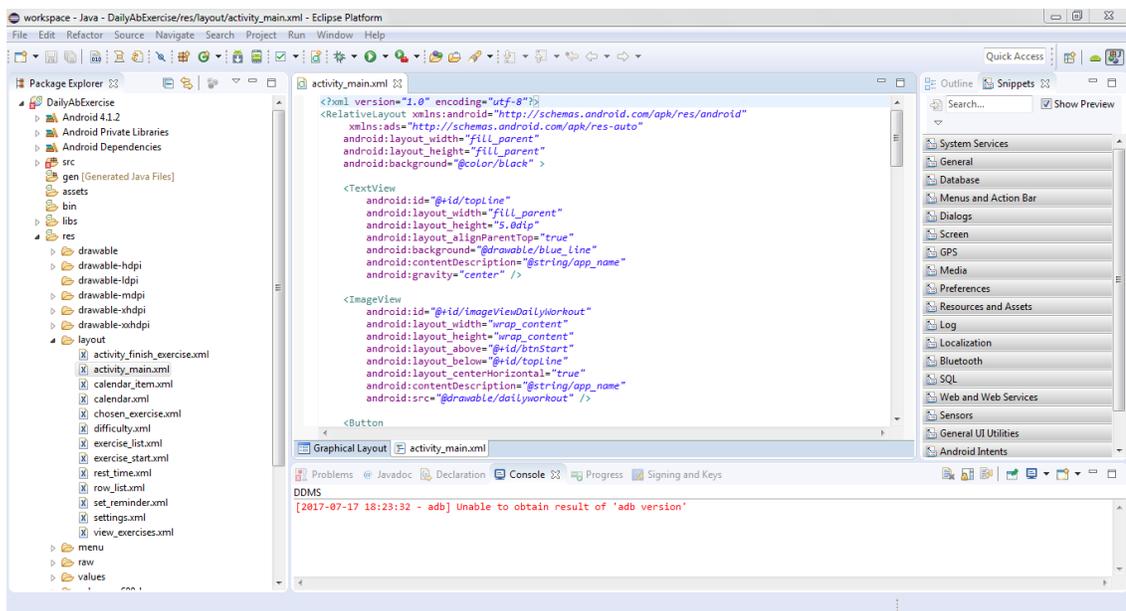
Content provaiders membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem memori, mengirim atau mengambil dari *database SQLite (MySQL)*. *Content provaiders*

menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*. Misalnya ketika menggunakan aplikasi yang membutuhkan peta (*Maps*) dan navigasi, mengakses data kontak, disinilah fungsi *content provider*.

Menurut Safaat (2015:58). “*Activity* adalah bagian penting dari keseluruhan siklus aplikasi, bagaimana *activity* dijalankan merupakan bagian penting dari model aplikasi.” *Activity* android dikelola dengan sistem yang dikenal dengan *activity stack*. Ketika suatu *activity start*, *activity* diletakkan pada *stack* (tumpukan) yang paling atas dari *activity-activity* yang sudah berjalan dibawahnya dan akan terus berada pada posisi atas *stack* sampai muncul *activity* yang baru. *Activity* memiliki empat keadaan antara lain: *activity (running)*, *pause*, *stopped*, dan *restart*. Berikut gambaran mengenai *activity* konsep

2.4. Pengujian Sistem

Menurut Safaat (2015:56) “Pengujian dilakukan pada fungsi-fungsi aplikasi, kemudian akan dilakukan pengamatan dari hasil pengujian tersebut untuk mengetahui kekurangan aplikasi dan kemudian dilakukan pengambilan keputusan.” Pengujian bisa dilakukan melalui *emulator 5554* standart bawaan android, jika *emulator* tersebut cukup berat sebagai *bulid target* alternatifnya menggunakan *emulator genymotion* 2.8.1. Atau langsung *build target* ke *handpone*, *tablet api 15* atau di atasnya sebagaimana pada awal pengujian sistem aplikasi android gambar berikut ini.



Gambar II. 4 Pengujian sistem

Sumber: Penulis

2.5. Peralatan Pendukung

Peralatan pendukung yang digunakan untuk membangun aplikasi My_Citra menggunakan windows7 spesifikasi *software aplikasi* pendukung sebagai berikut.

A. *jdk-8u60-windows-i586. (Aplication 185,160 KB)*

Ada dua produk utama dalam *platform* keluarga *Java SE: Java SE Runtime Environment (JRE)* dan *Java Development Kit (JDK)*.

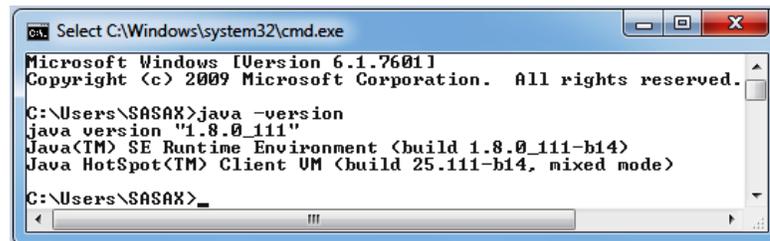
1) *Java Runtime Environment (JRE)*

JRE menyediakan perpustakaan, *Java Virtual Machine*, dan komponen lainnya untuk menjalankan applet dan aplikasi yang ditulis dalam bahasa pemrograman *Java*. Selain itu, dua teknologi penyebaran kunci adalah bagian dari JRE: *Java Plug-in*, yang memungkinkan untuk menjalankan applet di *browser* populer;

dan *Java Web Start*, yang menyebarkan aplikasi mandiri melalui jaringan. JRE tidak mengandung alat dan utilitas seperti *compiler* atau *debugger* untuk mengembangkan applet dan aplikasi.

B. *Java Development Kit (JDK)*

JDK merupakan superset dari JRE, dan berisi segala sesuatu yang ada di JRE, ditambah alat-alat seperti *compiler* dan *debugger* yang diperlukan untuk mengembangkan applet dan aplikasi. Diagram konseptual atas menggambarkan semua komponen teknologi dalam *platform Java SE* dan bagaimana mereka cocok bersama.



```
cmd Select C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\SASAK>java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) Client VM (build 25.111-b14, mixed mode)

C:\Users\SASAK>
```

Gambar II.5 *Java version*

Sumber: <http://www.oracle.com/technetwork/java/javase/tech/index.html>

C. **Android Studio**

Android Studio adalah sebuah IDE untuk pengembangan aplikasi diplatform Android. Sama seperti kombinasi antara Eclipse dan Android Developer Tools (ADT). Saat ini usia Android Studio masih tergolong muda, baru versi 0.2.3 (masih *early acces preview*). Berikut ini adalah sifat dari Android Studio :

1. *Multi-platform*
2. Solaris, AIX, HP-UX dan Mac OS X.
3. *Multi-language*

Android Studio dikembangkan dengan bahasa pemrograman Java, akan tetapi Android Studio mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

4. *Multi-role*

Selain sebagai IDE untuk pengembangan aplikasi, Android Studio digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.



Gambar II.6 *Android Studio v2.2.2*

Sumber: <https://developer.android.com/studio/intro/index.html>

D. *genymotion-2.8.1-vbox. (Application 157,663 KB)*

Genymotion tidak dapat dijalankan dari mesin virtual. Memang, dibutuhkan akses langsung ke perangkat keras untuk memberikan akselerasi *OpenGL*. Menjalankan

genymotion dari mesin virtual mencegah untuk langsung mengakses komponen *hardware*. Oleh karena itu, *genymotion* tidak sesuai ketika diinstal dari *VMWare*, *Parallels*, *Hyper-V* atau *VirtualPC*.

Untuk menjalankan perangkat virtual, harus menginstal *Oracle Vm Virtualbox*, *genymotion* mungkin bekerja dengan versi *Oracle VM Virtualbox*.



Gambar II.7 Emulator *genymotion* 2.8.1

Sumber: https://docs.genymotion.com/Content/01_Get_Started/Requirements.htm

E. VirtualBox

VirtualBox adalah sebuah software yang berfungsi untuk metode pembelajaran dalam menginstall OS (Operating System) secara virtual. Jadi pada intinya anda jika ingin belajar menginstall PC tidak perlu membongkar dan install ulang, cukup menggunakan VirtualBox anda pun bisa menginstall OS ke dalam PC. Hanya saja sifatnya virtual, tetapi tidak akan jauh beda dengan yang aslinya. Kata virtualisasi juga merujuk ke kamus Oxford (Convert "something" to a

computer-generated simulation of reality) yang artinya Mengubah atau mengkonversi ke bentuk simulasi dari bentuk yang real/nyata.



Gambar II.8 *VirtualBox*

Sumber : <http://www.anaktkj.net/2016/07/pengertian-manfaat-dan-fungsi.html>

Fungsi VirtualBox

1. Dapat mencoba operating system masih dalam tahap uji atau rilis
2. Dapat mencoba dan merasakan operating system yang berbeda dengan operating system yang utama
3. Dapat mencoba untuk membuat sebuah simulasi jaringan dan
4. Dapat juga untuk menguji dan mencoba simulasi system operating system security

Manfaat virtualBok

1. Menghemat uang dalam hal ini dapat di artikan anda tidak perlu membeli hardware atau computer baru untuk mencoba install ulang dan memakai banyak operating system.

2. Dapat menginstall operating beberapa operating system secara bersamaan tanpa harus memperainkan harddisk.

Dapat juga bermanfaat bagi orang yang awam untuk system pembelajaran tanpa harus memindah dan mengcopy data yang ada pada harddisk